"Express Mail" mailing label number:

EV304737593US

TRACKING COMMUNICATION FOR DETERMINING DEVICE STATES

Mark L. Wilkinson
Ronald J. Miller
Michael J. McDaniels

CROSS REFERENCE TO RELATED APPLICATIONS

[0002] This application claims the benefit of priority based on U.S. Provisional Patent Application Serial No. 60/415,290, filed October 1, 2002, entitled "System and Method for Detecting and Managing Network Intrusion," and naming Mark L. Wilkinson, Ronald J. Miller, and Michael J. McDaniels as the inventors. The above-referenced application is hereby incorporated by reference herein in its entirety.

[0003] This application is related to U.S. Patent Application Serial No. 10/262,321, filed October 1, 2002, entitled "System and Method for Managing Network Communications," and naming Mark L. Wilkinson, Ronald J. Miller, and Michael J. McDaniels as the inventors. The above-referenced application is hereby incorporated by reference herein in its entirety.

Field of Invention

[0004] This invention, in general, relates to a system and method for tracking communication between devices and inferring the state of at least one device based upon the observed communication.

Background of the Invention

[0005] Global networking of computers has greatly impacted business. As the number of

computers linked to networks grows, businesses increasingly rely on networks to interact. More and more people use email, websites, various file transfer methods, and remote office applications, among others, to facilitate business transactions and perform job related tasks.

[0006] These applications and uses still rely on early network addressing technologies and flow control protocols to transmit data packets across networks. For example, the Internet Protocol (IP) is an addressing protocol for referencing remote devices on a network. The protocol is implemented to include a packet header that contains bits representing an address of the source, an address of the target, and various other parameters associated with the packet. The Address Resolution Protocol (ARP) is used to reconcile physical addresses on local segments of a network with IP addresses. Other protocols are used for flow control including TCP and UDP. These protocols may be used to control the flow of packets across a network including subdividing and reassembling the packets. TCP also includes methods for verifying the arrival of a packet. Other protocols include ICMP, IPX, SPX, NetBios, and ARP, among others. Historically, these protocols were designed for use on a trusted network and as such do not include many security features. To address this problem, newer protocols are designed to include some security measures. However, at present, the global Internet and many local area networks predominantly use older protocols with various vulnerabilities.

[0007] Hackers and malfeasants take advantage of the weaknesses in these protocols to disrupt, infiltrate, or destroy networked devices. These attacks include worms, viruses, denial-of-service, and infiltration attacks, among others. Worms are self-replicating programs that infect computers. In some cases, these worms take advantage of the trusting relationships between computers to infiltrate network and send network data to the attacker. Viruses infect files and utilize vulnerabilities of programs that interpret the files to propagate. A virus may also function to erase data. Denial-of-services (DoS) attacks often limit the network activity of a target computer by inundating the target with requests or messages. In one example, an attacking computer or set of computers may send a plethora of low level pings to the target device. If the pings include a non-existent return address, the target machine could send a response message and pause over a timeout period for a response. In attempting to respond to the pings, the machine effectively denies network access to other applications.

[0008] Infiltrating attacks often circumvent password security and gain access to files.

Once the attacker has access, they may steal private information such as credit card or social security numbers. Moreover, they may damage valuable data, install a worm or spying program, or install programs to utilize computational capacity.

[0009] The FBI reports that millions of dollars are lost each year as a result of these attacks. In the "2002 Computer Crimes and Security Survey," as much as 90% of the Fortune 2000 companies reported breaches in computer security. According to the survey, each successful attack cost corporations an average of $2.1 Million. The losses include lost data, employee time used in recovering data, delays in existing projects, and damage to equipment. Fifty five percent of the companies surveys reported denial-of-service attacks, 70% reported infiltration and vandalism attacks, and twelve percent reported theft of transaction information.

[0010] Hackers use various tools and methodologies to discover vulnerable devices and interact with them. These tools include address scanners, port scanners, worms, and packet formulation programs, among others. For example, a hacker may send reconnaissance packets to a local network segment in search of a computer or device. Once a device is found, the hacker may scan the ports on the device in search of a vulnerable port.

[0011] Several approaches exist for protection against hackers. Typically, these protections are defensive shield-like methods. The most common are firewalls, intrusion detection systems (IDS), and anti-virus software. Firewalls are devices typically placed as shields between a local network and the global network. Firewalls are the most common form of network protection. They perform their function by limiting communication between the local network and global network in accordance with various filters and rules. Typically, network traffic is either blocked or permitted based on rules regarding protocols, addressing, and port number. These filters are infrequently changed and can unintentionally encumber certain permissible network traffic while permitting unwanted traffic.

[0012] Intrusion detection systems detect intrusions or attacks and report these attacks to network security. The systems predominantly use packet signatures to evaluate network packets. However, these systems have been shown to be unreliable as they can generate false positive results. Often, the systems collapse under the weight of the data they collect. Further, these systems may not detect packets with signatures that are not found in their signature database, resulting in false negatives as well. Moreover, these systems often

present the data to network security in a format that prevents timely response to threats.

[0013] Similarly, anti-virus software typically relies on file signatures to detect viruses. As such, frequent updates are required to maintain a current database of virus signatures. If an undocumented virus enters the network, the anti-virus software will likely fail.

[0014] Many network security systems suffer from deficiencies in detecting and preventing attacks on a network. Many other problems and disadvantages of the prior art will become apparent to one skilled in the art of networks security systems after comparing such prior art with the present invention as described herein.

Summary of the Invention

[0015] Aspects of the invention may be found in a system and method for tracking communication for determining device states. The method can include observing communication between devices and inferring a respective state of at least one of the devices based upon the communication observed. The inferring the respective state of a device can be performed without sending a packet to the device, without participating in the communication with the device, and only by listening to the communication with the device.

[0016] The method can further include setting a designation for a first device to a threat when the first device receives a packet and the respective state of the first device is unfulfilled. The method can further include changing the designation for the first device to a non-threat when subsequent communication initiated by the first device does not violate a rule for the communication.

[0017] The method can include setting a designation for a first device to a possible threat when communication is initiated by the first device, and the communication initiated by the first device violates a rule. The method can further include changing the designation for the first device to a non-threat when subsequent communication initiated by the first device does not violate rules for the communication. The method can include setting a designation for a first device to a possible threat based upon a packet configuration for a packet sent by the first device as part of the communication.

[0018] Various states of the devices include unknown, used, unfulfilled, virtual, omitted, and automatic. The respective state of a device is determined to be unknown when the

observation shows that the device fails to respond to communication sent to the first device. The respective state of the device is determined to be unfulfilled when an address resolution protocol request comprising a destination address for the device is observed, and the device does not respond to the address resolution protocol request prior to expiration of a time limit. The respective state of a device is determined to be used when the observation indicates that the device performs one of sending and receiving a packet. In addition, the respective state for the device is determined to be used when the observation shows that the device received a packet when its respective state was unfulfilled, and the device sent a reply to the packet within a time limit.

[0019] The respective state of a device is determined to be virtual when the observation shows that the device received a packet when its respective state was unfulfilled, and the device did not send a reply to the packet within a time limit. The respective state of the first device is determined to be automatic when an automatic reply is programmed to be sent to a second address when the first address receives a packet from the second address. The respective state of the device is determined to be omitted when the device has been programmed such that communication with the first address is omitted from observation.

[0020] In one embodiment, the method includes initializing the respective state of at least one device to unknown prior to beginning the observation. The plurality of devices can communicate via a segment of a network. In one embodiment, the method includes maintaining the respective state for one device in a storage area. In one embodiment, the method includes storing information about at least one packet of a plurality of packets communicated between the devices.

[0021] Additional aspects of the invention may be found in a system for tracking communication for determining device states. The system may be a computational device including a processor or network interface and memory or computer-readable medium, among others. The device may or may not include a user interface. Further, the device may have various data and instructions associated with various methods for tracking .communication for determining device states These data may include an ARP request queue, an IP state table, a frequency table, an ARP table, a watch list, a threat list, a synthetic physical address table, and a communications stream table, among others. The device may also include instructions for evaluating packet reconnaissance rules, behavioral rules and

other rules, among others. Further, the system may include software or computer interpretable instructions for performing various methods associated with maintaining the data in the tables and collecting the data for the tables.

[0022] As such, a system and method for tracking communication for determining device states are described. Other aspects, advantages and novel features of the present invention will become apparent from the detailed description of the invention when considered in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0023] For a more complete understanding of the present invention and advantages thereof, reference is now made to the following description taken in conjunction with the accompanying drawings in which like reference numbers indicate like features and wherein:

[0024] FIGURE 1 is a schematic block diagram depicting a network and network devices according to the invention;

[0025] FIGURE 2 is a schematic block diagram depicting an exemplary embodiment of a network as seen in FIGURE 1;

[0026] FIGURE 3A is a block flow diagram depicting an exemplary method for use by the system as seen in FIGURE 1;

[0027] FIGURE 3B is a schematic block diagram depicting an exemplary embodiment of network protocols in use on the system as seen in FIGURE 1;

[0028] FIGURE 4 is a block flow diagram depicting an exemplary method for use by the system as seen in FIGURE 1;

[0029] FIGURE 5 is a block flow diagram depicting an exemplary method for use by the system as seen in FIGURE 1;

[0030] FIGURE 6 is a block flow diagram depicting an exemplary method for use by the system as seen in FIGURE 1;

[0031] FIGURE 7 is a block flow diagram depicting an exemplary method for use by the

system as seen in FIGURE 1;

[0032] FIGURE 8 is a block flow diagram depicting an exemplary method for use by the system as seen in FIGURE 1;

[0033] FIGURE 9 is a block flow diagram depicting an exemplary method for use by the system as seen in FIGURE 1;

[0034] FIGURE 10 is a block flow diagram depicting an exemplary method for use by the system as seen in FIGURE 1;

[0035] FIGURE 11 is a block flow diagram depicting an exemplary method for use by the system as seen in FIGURE 1;

[0036] FIGURE 12 is a schematic depicting an exemplary embodiment of an ARP queue as used in the method as seen in FIGURES 5-11;

[0037] FIGURE 13 is a block flow diagram depicting an exemplary method for use in evaluating the ARP queue as seen in FIGURE 12;

[0038] FIGURE 14 is an exemplary embodiment of a watch list as used in the methods as seen in FIGURES 5-11;

[0039] FIGURE 15 is a block flow diagram depicting an exemplary method for use in evaluating the watch list as seen in FIGURE 14;

[0040] FIGURE 16 is a schematic depicting an exemplary embodiment of a multidimensional frequency table for use in the methods of FIGURES 5-11;

[0041] FIGURE 17 is a schematic depicting an IP state table for use in the methods as seen in FIGURES 5-11;

[0042] FIGURE 18 is a schematic diagram depicting an table for use in the method as seen in FIGURES 5-11;

[0043] FIGURE 19 is a schematic of an exemplary embodiment of a message header for use in the methods as seen in FIGURES 5-11;

[0044] FIGURE 20 is a schematic of an exemplary embodiment of a communications stream table for use in the methods as seen in FIGURES 5-11;

[0045] FIGURE 21 is a block diagram depicting an exemplary embodiment of a network security device for use in the system as seen in FIGURE 1;

[0046] FIGURE 22 is a block flow diagram depicting an exemplary method for use by the system as seen in FIGURE 1;

[0047] FIGURE 23 is a block flow diagram depicting an exemplary method for use by the system as seen in FIGURE 1;

[0048] FIGURE 24 is a block flow diagram depicting an exemplary method for use by the system as seen in FIGURE 1;

[0049] FIGURE 25 is a block flow diagram depicting an exemplary method for use in the system as seen in FIGURE 1;

[0050] FIGURES 26A and 26B are block flow diagram depicting exemplary methods for use in the system as seen in FIGURE 1; and

[0051] FIGURE 27 is a schematic diagram depicting an exemplary embodiment of a network as seen in FIGURE 1.

DETAILED DESCRIPTION

[0052] In attempting to contact and infiltrate networks, attackers or programs implemented by the attackers act with characteristic behaviors to determine the address of computers on local segments and communicate with them. These behaviors may be used, separately, or in combination with packet signatures and filters, to identify the attackers as threats. Once identified, communication between the network and the attacker may be controlled, preventing further damage. One method is to deceive the attacker, preventing them from either perceiving the existence of a machine or redirecting their communication to an alternative device such as a security device or sacrificial computer.

[0053] FIGURE 1 is a schematic block diagram depicting a network according to the invention. An area network 14 is connected to a global network 12. A firewall 16 may or may not be placed between the global network 12 and the area network 14. Connected to the area network 14 may be routers 20, switches 22, security devices 24, servers 26, workstations 28, portable devices 30, and gateway devices 34, among others. The network 14 has either a security device 24 or some other computational device that acts to detect attacks on the area network 14 and mask other devices attached to the area network 14 from the attacker. The attacker may be an attacker 18 connected to the global network 12 or an attacker 32 connected to the area network 14. However, these components may or may not be included in the system, and may be together, separate, or in various combinations, among others.

[0054] The global network 12 may take various forms and may communicate with various protocols including IP, TCP, UDP, ICMP, HTTP, and FTP, among others.

[0055] Area network 14 may take various forms including Ethernet, Wireless Ethernet, Token rings, Apple Talk, or various combinations of these, among others. In one exemplary embodiment, the area network 14 may be an Ethernet network that resolves logical and physical network addresses using an address resolution protocol (ARP). The ARP resolves the addresses between Internet Protocol (IP) addresses and physical address such as media access control (MAC) addresses.

[0056] The security device 24 or some other computational device such as a server 26, workstation 28, routers 20, switches 22, or gateways 34, among others, may function to detect attacks on the area network 14. Singly or in combination these devices may hold and

compile a list of devices, MAC addresses, or source IP addresses of devices that represent a threat to the system. Using this list, the device or devices may capture packets, compare the MAC address, source IP address, or target IP address, with known threats to the system, and take steps to control or prevent communication with vulnerable devices.

[0057] For example, the device may create ARP packets with synthetic hardware addresses associated with the IP addresses of either local devices or attacking devices. In this manner, the ARP tables may be altered, causing packets to be sent across the network to physical addresses other than those targeted by the communication.

[0058] For example, the synthetic hardware addresses may be MAC addresses that are not in use by devices on the area network 14. Alternately, the synthetic hardware address may be the address of a sacrificial computer, defense system, or security system, among others.

[0059] FIGURE 2 is a schematic block diagram depicting an exemplary embodiment of the system as seen in FIGURE 1. The router 38 may or may not be connected to the global network 37 through a wide area network 39. The router 38 routes traffic from the global network 37 or wide area network 39 to various devices on a local area network or a local segment of a network 48. The devices on the segment may include workstations 40 and 46, a server 42, a gateway 41, and a security device 44. However, various embodiments of network segments may be envisaged.

[0060] In this exemplary embodiment, attacker 36 may send network packets through a global network 37 to router 38. Router 38 may then forward the network packet to the local network 48 where the addressed device receives the communication. Security device 44 may capture these packets, determine whether the packets were sent by an IP address of a device of interest or that represents a threat, and act to control communications between the attacker 36 and the devices on the local segment 48.

[0061] In one exemplary embodiment, a security device 44 may detect an attack on the local segment 48. The device may then create ARP packets that include a synthetic physical address, and send those ARP reply packets to a gateway device or other computational device, effectively altering the ARP table in that device.

[0062] FIGURE 3 is a block flow diagram depicting a generic method for detecting network incursion. The method includes capturing network packets from the local segment. From these packets tables associated with the state of IP address, the nature of the packets and communications streams may be established. This data may be compared to various rules to determine if an attack is occurring. In the event that an attack is occurring, the system may set a flag or parameter indicating that occurrence and notify personnel or implement a defense mechanism.

[0063] Various methods for identifying threats and therefore addresses of devices that represent these threats may be envisaged. One exemplary embodiment is seen in the block flow diagram of FIGURE 4.

[0064] The method 90 begins with the capturing of a network packet from the local segment as seen in a block 92. This packet may be a network packet comprising data for various protocols including ARP, TCP, IP, HTTP, UDP, or FTP, among others. The method may then decode the packet as seen in a block 94. With the decoded packet, system determines whether the packet represents is of interest or a known threat and may implement a control mechanism such as adjusting an ARP table in response. For example, the system may compare the source or target IP address to a list of known threatening devices. However, alternate methods may be used in determining the threat.

[0065] The system may then process various parts of the packet or packet formats in the appropriate manner. For example, the system may process ARP packets as seen in a block 98 and TCP/IP packets as seen in a block 100. However, the method may be configured to process packets using a variety networking and data linking protocols.

[0066] The system may then compare the processed information and packet to a set of reconnaissance rules as seen in a block 102. Multiple reconnaissance rules can be used. For example, determining if the packet strictly conforms to the protocol specification for size and configuration, identifying if the packet flags represent any illegal combinations used to circumvent firewalls, or determining if the packet continues a pattern of packets that cumulatively represents a reconnaissance event. In this manner, the system may determine whether a source computer is behaving in an appropriate manner, and if not, may place the address of the computer on a watch list.

[0067] Next, the system may check for target violations as seen in a block 104. These violations may include a packet addressed for an IP address known not to be in use, a packet addressed to a port that is known to be closed, or an ARP request sent to an IP address that does not provide a corresponding reply.

[0068] The system may also update or add a communication stream to a communication stream table as seen in 106. The communications stream table may be one or more tables recording information associated with the communications protocol used by the packet, the addresses of the devices associated with the packet, the direction of the packet, and the time of the last packet sent, among others. For example, the system may maintain several tables, each uniquely associated with a communications protocol. These tables may then track interactions with devices on the area network. In this manner, the system may track a communication stream and determine whether systems or devices are behaving appropriately.

[0069] As seen in a block 108, the system may compare the behavior of various devices to a set of behavior rules to determine whether a device as represented by its IP address is acting as a trusted device should. For example, a device scanning multiple IP addresses may represent a threat. Or, a device may begin receiving traffic on a port that has historically been closed. Or, the device may initiate communication on ports representing services that are not installed on the device. In all cases, the device would be added to a watch list. A more detailed embodiment of this method may be seen in FIGURES 5-11.

[0070] As seen in FIGURE 5, the system may capture a packet 112 and decode the packet as seen in a block 114. The system may first test to determine whether a threat exists to the system. This test may be implemented as a flag in a data set or a test for IP addresses in a list of known threats, or in various other manners. For example, if a device with an IP address known to be a threat has active communications streams in one or more communications streams tables, the system may change a flag or variable to represent the existence of a threat condition. Over time, if communication from the threatening device ceases, the system may return to a state of no threat. If no threats exist, then the system may move to the processing of ARP packets as denoted by a block A.

[0071] However, if a threat exists, the system may test the source IP address of the packet as seen in a block 118, the target IP address as seen in a block 120 and the target physical address as seen in a block 122. If the target IP address or source IP address are determined to

represent threats, the system may invoke a defense mechanism as seen in block 124. This defense mechanism may include managing communications through MAC layer routing and filtering. Additional defense methods may be seen in FIGURES 25, 26A, and 26B.

[0072] To determine whether the addresses represent a threat, the system may compare the addresses to a list of addresses of known threats. Similarly, if the target physical address is synthetic the system may invoke a defense as seen in a block 124. To determine whether the physical address is synthetic, the system may compare the address to a list of known synthetic addresses, or a list of known addresses in use on a local segment. If no threat is found and the physical address is not synthetic the system may continue as seen in a block A.

[0073] In an alternate embodiment, the system may determine whether the packet with the synthetic physical address represents a threat. If the packet is non-threatening, the packet may be reformulated and sent to the intended device.

[0074] FIGURE 6 depicts a processing of packets that utilize the address resolution protocol (ARP). The address resolution protocol includes protocols for several message types. Two of the types include requests and replies. Generally, a request from one device has a corresponding reply from another device. A lack of a corresponding relationship may indicate the presence of an attack.

[0075] Once the packet is tested for threats or synthetic addresses, it may be tested to determine whether the packet uses ARP. If the packet does not use ARP, a system may move on to the processing of TCP/IP packets. However, if the packet is an ARP packet, the packet may be tested for various types of ARP packets, and processed accordingly. For example, if the packet is an ARP request as seen in a block 184, the packet may be further tested to determine whether the packet is gratuitous as seen in a block 136. A gratuitous ARP request is typically used to announce to a network the presence of a new device and its address. If the packet is a gratuitous ARP, the system may continue on to the processing of TCP/IP packets.

[0076] However, if the packet is not gratuitous, the system may test to determine whether an identical ARP request is found in an ARP queue as seen in a block 138. If an identical request is found, the packet may continue on to TCP/IP processing. However, if the packet is not in the queue, the ARP request may be added to the ARP request queue as seen in a block

140, and the state of target IP may be changed to "unfulfilled" in an IP state table. In this manner, state data associated with requests may be stored while waiting for an expected reply.

[0077] If the packet is an ARP reply as seen in a block 144, the system may test for a corresponding ARP request in a queue as seen in a block 146 and remove the corresponding ARP request from the queue as seen in a block 148. In each case, the packet may be forwarded for TCP/IP processing as denoted in block C. An exemplary embodiment of an ARP request queue is seen and discussed in relation to FIGURE 12. An exemplary embodiment of an IP state table is seen and discussed in relation to FIGURE 17.

[0078] Once ARP packets have been processed, or if the packets are determined not to be ARP packets, then they may be forwarded to TCP/IP processing as shown in FIGURE 7. The packet may first be tested to determine whether the packet is a TCP/IP packet. If it is not, it may be forwarded to processing under alternate protocols or a subsequent packet may be captured and the method re-started. For example, this may be the case for networks with IPX/SPX communications.

[0079] If the packet is a TCP/IP packet, the source IP address is sought in a frequency table as seen in a block 154. If the source IP address is not found in the frequency table it may be added as seen in a block 156. In either case, the system tests the destination IP address to determine whether it is in the frequency table. If the destination IP address is not in the frequency table, it may be added. In either case, the system may update packet frequency values in accordance with the frequency table as seen in a block 162. An exemplary embodiment of a frequency table is seen and discussed in relation to FIGURE 16.

[0080] FIGURE 8 is an exemplary method for determining whether a packet is designed for reconnaissance in a local network segment. For each rule in a reconnaissance rule list, as seen in a block 172, a system may test a packet to determine whether it violates the rule as seen in a block 174. If the packet violates a rule, a system may note the rule and place the IP source on a watch list as seen in a block 178. A system may then continue to test rules as seen in a loop from block 180 to 172. Once all the rules are tested, the system may continue on as seen in block E. Some examples of common reconnaissance rules are: packets with parameters that violate the protocol definition, packets with inappropriate flag configurations, or packets with unusual sequencing or fragmentation.

[0081] FIGURE 9 is an exemplary method for testing a target violation. This method tests to determine if a packet has a destination address that is not in use, a port that is known to be closed, or an ARP request with an IP address that does not provide a corresponding reply. The system tests the packet to determine whether it is an ARP request as seen in a block 192. If the packet is an ARP request, a system checks the target IP state to determine whether it is unfulfilled, as seen in a block 194. If it is, the system continues as seen in a block F. However, if the target IP state is not unfulfilled, the system notes a rule violation as denoted by the block 200 and places the source IP address on a watch list as seen in a block 202. If the packet is not an ARP request, the system tests the target IP address of the packet to determine whether it is in use as seen in a block 196. If the target IP address is in use, the system checks to see if the target port on the target device is known to be open as seen in a block 198. If the target IP address is not in use or the target port is not known to be open, the system notes a rule violation as seen in a block 200, and places the source IP address of the packet on the watch list. The system then continues as denoted by a block F.

[0082] FIGURE 10 is an exemplary method for updating a communication stream table from block F. The packet is tested to determine whether it is included in the communication stream table as seen in a block 212. If it is not, it is added to the table as seen in a block 214. If it is, the table is updated with the communication stream time and direction, as seen in a block 216. The system then continues as seen in a block G.

[0083] Once the communications tables are updated, the system may check for behavioral rule violations. FIGURE 11 is an exemplary method for testing whether a behavior rule is violated. For each behavior rule, the system checks the frequency data in the frequency table to determine whether a cell, row, or column in a multi-dimensional array has violated a behavioral rule boundary as seen in a block 234. If the boundary has been violated, the rule is noted as seen in a block 236 and the source IP address is placed in a watch list as seen in a block 238. Once the rules have been tested, the system returns as seen in a block H and continues to collect a subsequent packet, decode it, and process it in the manner seen in FIGURES 5-11.

[0084] FIGURE 12 is an exemplary embodiment of an ARP request queue. The ARP request queue may include a source IP address, a target IP address, a source MAC address, and the time associated with an ARP request. If a corresponding reply to the request is not

received within a specified programmable period of time, the ARP request may have been initiated by a device that represents a threat.

[0085] FIGURE 13 is an exemplary method in which the time is compared to a set threat limit as seen in a block 254. If the time exceeds the threat limit, the source IP address is placed on a watch list as seen in a block 256. This step may be repeated for each item in the queue as seen in a block 252.

[0086] FIGURE 14 represents an exemplary embodiment of a watch list. The watch list may include a listing of potentially threatening devices. The watch list, for example, may contain a source IP address, a target IP address, a source physical address, the time the packet was sent, and the denotation of a rule that was violated, among others.

[0087] FIGURE 15 is an exemplary method for determining whether items in a watch list represent a threat. For each item in the watch list, the system may determine whether it is necessary to elevate or escalate the item to a threat as seen in a block 274. This decision may be based on the type or frequency of rule violations, or various combinations of rule violations and frequency violations. If it is necessary to escalate the watch list to a threat, the system may add the source IP address to a threat list as seen in a block 276. However, various methods may be envisaged.

[0088] Conversely, it may become apparent over time that a source IP address does not constitute a threat, in which case it can be de-escalated. For example, a typographical error may have been made in a destination address, without the intent or function of generating a threat. In such an instance, de-escalation would be the appropriate way to return the system to a normal condition.

[0089] FIGURE 16 represents an exemplary embodiment of a frequency table. In this exemplary embodiment, a multi-dimensional table is shown with a vertical axis of IP addresses, a horizontal axis of time, and another depth axis. The vertical axis may be a listing of IP addresses. In this exemplary embodiment the listing of IP addresses includes all source addresses of packets.

[0090] The time axis may represent bins of varying periods. For example, these periods may represent the time since the last packet associated with the IP address or ongoing time.

The number of bins may be set by a user of the system and the size or period represented by the bins may be set and varied along the axis.

[0091] Looking at this two-dimensional face, the IP addresses and time, the number of packets sent by the IP address may be tallied over time to produce a frequency table of packets associated with the IP address. Various methods may be used to normalize the frequency table or adjust the values in the various time bins over time or in association with the arrival or transmission of subsequent packets. In one exemplary embodiment, each bin representing a period less than the period of the most recent packet may be decremented uniformly or with some distribution.

[0092] In the multi-dimensional case, other axes may be represented by packet size, destination IP addresses, protocols, message types, and other characteristics. Various behavioral rules may then be established that identify threats by exceeding the boundaries assigned to various cells, columns, or rows. For example, if a single source as represented by its IP address were to frequently send packets into the network to multiple unused IP addresses, that source may be identified as a potential threat seeking the addresses of devices. In other example, if various sources flood a single IP address with messages, it may be determined that a denial of services attack is occurring.

[0093] FIGURE 17 represents an IP state table. Recall that when an ARP request is identified and added to the ARP request table, the IP state of the target IP address or device is set to "unfulfilled" in an IP state table. Traditionally, an IP address may be in use or not in use. This system, however, tracks various additional states including omitted, automatic, used, unfulfilled, virtual, and unknown. The system uses the IP state table to monitor IP addresses associated with the local network segment and those addresses on the threat list. The IP state table provides the system with a method of monitoring the state of IP addresses on a dynamically changing network segment.

[0094] In the case of an ARP request sent to a target IP address, the system sets the state of the target IP address to unfulfilled. If a reply to the ARP request is not received by the source IP address within a given period of time, the target IP address state may be set to virtual. It is possible that the target address is not in use. If a packet is sent from an IP address that is marked as unknown, the system attempts to ascertain the true state of the IP address through various active and passive means. When the true state of the IP address is

determined, the state table is updated with the appropriate state value.

[0095] Since communication between devices is perceived through packets, the state of the network may be tracked through the IP State table. FIGURE 18 depicts a conceptual chart of communications. An IP address may be in use or not in use. A packet may be sent from a source IP address known to be in use to a destination IP address. If the destination IP address is not in use, the sending of the packet to that destination address may be a first clue in perceiving network incursion. Similarly, a sending of a packet from a source known not to be in use may indicate address spoofing by an intruder.

[0096] However, a single packet such as an ARP request does not indicate whether the destination IP address is in use. An ARP reply would indicate the use of the destination address. Failure to reply may indicate that the destination address is an unused IP address. However, a device associated with the destination address may have been busy. As such, the devices may be categorized in the state table in categories indicative of knowledge or expectation obtained from observation of other communication with those devices.

[0097] Turning to FIGURE 19, an ARP reply packet 290 is depicted. In this exemplary embodiment, the ARP reply packet 290 is depicted as having a source IP address 292 with an associated hardware address 294. The associated hardware address is, in this example, a synthetic hardware address. The message 290 also contains a destination IP address 296 and a hardware address 298 associated with the destination IP address 296. If such a message were sent on a local network segment, the ARP table associated with the destination IP address 296 and hardware address 298 would be adjusted. Any subsequent communications sent to the source IP address from the device associated with the destination address would be sent to an incorrect hardware address, preventing communications with the source device represented in the ARP reply packet 290. The destination device may be a server, gateway, workstation, or other network device. In this manner, communications may be controlled between attacking devices and devices on the local segment of the network.

[0098] Turning to the next figure, FIGURE 20 depicts an exemplary embodiment of a communications stream table. The communications stream table is one or more tables tracking communications between devices. In one exemplary embodiment, the communications stream tables are tables that each track communications associated with a single protocol. For example, tables may be used to track ICMP, TCP, UDP, and ARP

communications. The tables may be associated with the addresses of two devices associated with the number and frequency of data, direction, statistical information, and time of last packet. The communication streams tables may be periodically polled to aid in setting the state of IP addresses in the IP state table. For example, if the age of a communications stream exceeds some value, the system may attempt to contact the device associated with the IP address to determine whether it is active or the address is in use. If the device does not respond, the device may be removed from the IP state table. Furthermore, the communications stream tables may be updated if a communications stream becomes stale.

[0099] FIGURE 21 is an exemplary embodiment of a security device for performing the functionality described in the various methods herein. The security device 310 may include a processor 312, a network interface 314, a memory 316, and a user interface 340. The device 310 may also include an ARP request queue 318, an IP state table 320, a frequency table 322, an ARP table 324, a watch list 326, a threat list 328, packet reconnaissance rules 330, behavior rules 332, a MAC address table 334, one or more communications stream tables 336, and various instructions 338, among others. However, each of these elements may or may not be included together, separately, or in various combinations, among others. For example, the data and instructions may reside on a single device or on various devices, among others.

[0100] The processor 312 may take various forms including various microprocessors, computational circuitries, and controllers, among others. The processor 312 may interpret various instructions or data and function accordingly.

[0101] A network interface 314 may take various forms. These forms may include an Ethernet NIC, a serial cable, a USB port, an AppleTalk connection, and a wireless Ethernet connection, among others. The network interface may function to aid in capturing network packets, sending appropriate messages across the network, and communications with other devices, among others.

[0102] The memory 316 may take various forms including ROM, RAM, flash memory, disc drives, floppy drives, CD ROM's, and DVD ROM's, among others.

[0103] The system 310 may or may not include a user interface 340. This user interface may take various forms including a hand-held device, a keyboard, a monitor, a mouse, and

remote access interfaces, among others.

[0104] The ARP request queue 318 is a listing of ARP requests and the time they were sent. The system attempts to match ARP requests with a corresponding ARP reply. If no match is found over a given period of time, a rule may be violated and the source IP of the ARP request may be placed on a watch list. The table may take various forms including a database file, a tab delimited file, a spreadsheet, a text file, a data file, among others.

[0105] The IP state table 320 may include a listing of IP addresses and an associated state. Generally the state is either active or inactive. However, the system applies varying states to the IP address including omitted, automatic, used, unfulfilled, virtual, and unknown, among others. The table may take various forms including a database file, a tab delimited file, a spreadsheet, a text file, or a data file, among others.

[0106] The frequency table 322 may take various forms, including that shown in FIGURE 16. The frequency table may store data associated with an IP address and the frequency of packet delivery over time. Further, the table may be structured as a multi-dimensional data set with other axes including protocols, packet size, communications type, and destination address, among others. The frequency table may take various forms including a database file, a tab delimited file, a spreadsheet, a text file, a data file, among others.

[0107] The ARP table 324 is typically a listing of IP addresses and an associated physical address on the network. Generally, most devices connected to the network maintain an ARP table. This system may communicate with various devices on a network to adjust values maintained in the various ARP tables.

[0108] The watch list 326 may take the form as seen in FIGURE 14. The watch list may include a listing of IP addresses that have violated various rules. IP addresses listed in the watch list may be elevated and identified as potential threats, and placed in a threat list 328. The threat list may be a listing of IP addresses associated with devices that are known to be threats to the system. This list may be used to determine which packets entering the local network segment represent a threat and to prevent those packets from accessing devices on the local segments.

[0109] A device 310 may include packet reconnaissance rules 330. These rules may be

used to determine from the packet information whether the packet is designed to reconnoiter on the local segment. If the packet violates these rules, the source IP address for the packet may be placed on the watch list.

[0110] The behavioral rules 332 may be rules, boundaries, or thresholds that are compared to frequency table 322. If cells, columns, or rows violate these boundaries or thresholds, the IP address associated with the violation may be marked as a potential threat and placed on a watch list or the threat list 328.

[0111] The system may also include a table of synthetic MAC addresses. Alternately, the system may include a version of an ARP table associating IP address with real physical addresses, or a combination of both synthetic MAC addresses and real physical addresses. It should be understood through this description of an illustrative embodiment, that although the use of tables is mentioned here, the present invention may use other types of persistent, logically addressable storage mechanisms for the different types of addresses. In either case, the table may be used to determine whether an address is a synthetic address and be used in re-creating or re-structuring the packets for delivery to the appropriate physical device.

[0112] The Communications stream table or tables 336 may take various forms including a database file, a tab delimited file, a spreadsheet, a text file, a data file, among others. The communications stream tables 336 may track communications between devices as discussed in relationship to FIGURE 20.

[0113] The instructions 338 may include various operating instructions and computer-implemented instructions for implementing the methods herein, among others. These instructions may take the form of interpretive instructions, programs, and additional data, among others.

[0114] However, these elements may or may not be included together, separately, or in various combinations, among others. For example, various devices may be combined to function and store the data and instructions described above.

[0115] FIGURE 22 is another exemplary method for perceiving threats to the system. Hereto the system captures the packet as seen in a block 354. The system determines whether the packet is an ARP packet as seen in a block 356. The packet may be an ARP

request or an ARP reply as seen in a block 358. From this it may be determined whether the source IP address is a threat as seen in a block 362, or not a threat as seen in a block 360.

[0116] Further, the packet may be tested to determine whether it is a TCP/IP packet as seen in a block 364. If it is not a TCP/IP packet, it may be ignored as seen in a block 366 or forwarded for processing in accordance with other packet protocols. If the packet is a TCP/IP packet, a destination IP address may be tested as seen in a block 368. If the destination IP address is not in use, the source IP address may represent a threat as seen in a block 370. If the destination IP address is in use, the packet may be subjected to further tests. For example, the packet may be tested to determine whether it represents a reconnaissance packet as seen in a block 372. If it is a reconnaissance packet, the source IP address may be a threat as seen in a block 374.

[0117] Further, the packet may be tested to determine whether it violates a frequency rule as seen in a block 376 or a target port frequency rule as seen in a block 380. In either case, a violation of a rule may indicate that the source IP address is a threat as seen in blocks 378 and 382, respectively. Further, the packet may be tested to determine whether it violates the aggregate rules as seen in a block 384. If it does violate these rules, it may represent a threat as seen in a block 386. However, if the packet does not violate the rules, it may not represent a threat as seen in a block 388.

[0118] FIGURE 23 is an exemplary method for creating and maintaining an IP address state table. Hereto the system may capture a packet as seen in a block 394. The system may test to determine whether the source IP state is known. If it is not known, the system may set the source IP state as seen in a block 398 and note the time as seen in a block 400. If it is known, the system may test to determine whether the destination IP address state is known. If that state is not known, the destination IP address state may be set as seen in a block 406, and the time noted, as seen in a block 408. Further, the system may test the packet configuration to determine whether the source IP state requires change as seen in a block 412. If the source IP state is to be changed, the system may change these source IP states as seen in a block 414 and note the time as seen in a block 416. In either case, the system may further test to determine whether the destination IP state requires change as seen in a block 418. If change is required, the system may change the state of the target IP address as seen in a block 420 and note the time as seen in a block 422.

[0119] FIGURE 24 represents adjustment of an IP address state. In this case, the system checks the time as seen in a block 434. The system determines whether there is an appropriate time to check the IP state as seen in a block 436. If it is not, the system does nothing as seen in a block 438. If it is time to check the states, the system checks each entry on an IP state table as seen in a block 440. If enough time has elapsed, as seen in a block 442, the system adjusts the IP state as seen in a block 446.

[0120] FIGURE 25 is an exemplary embodiment of a defense method as described in relation to FIGURE 5. The system may accept a packet as seen in a block 472. The system may test the source IP address of the packet to determine whether it represents a threat. If the source IP address does not represent a threat, the system may accept a subsequent packet. If the source IP address does represent a threat, the system may test the packet for various protocols and packet types as seen in blocks 476, 480 and 484. For example, if the packet as an ARP request from a known threat, the system may send an appropriately crafted ARP reply. The reply fools an attacking computer into believing that a computer exists at the target IP address and sets the stage to allow for future connections.

[0121] If the packet is an ICMP echo request as seen in a block 480, the system may send an appropriately crafted ICMP echo reply as seen in a block 482. In this case, an attacker may be performing reconnaissance. An appropriate response makes the attacking computer believe a real computer exists at the target IP address.

[0122] If the packet is a TCP packet as seen in a block 184, the system may test for various types of TCP packets. For example, the system may test to determine whether the TCP packet is a SYN request as seen in a block 486. If it is, the system may create an appropriate acknowledgement response as seen in a block 488. This acknowledgement response may include the TCP window size set to zero and the data of payload size set to a very small number as seen in blocks 490 and 492. The TCP SYN request indicates that an attacker may be trying to connect the first time. By sending the suggested response, the attacker's computer is provided with parameters requiring it to do extra work sending small packets and effectively occupying computational cycles. If enough threads of such TCP packets are sent out, communications to the attacker's computer may effectively be slowed.

[0123] If the TCP packet is a window probe packet as seen in 496, the attacker may be trying to increase the speed of their attack by requesting a verification of a maximum window

size. Here, to, an appropriate TCP response packet may be crafted as seen in a block 498. Again, the TCP window size may be set to zero, and the window probe response packet be sent as seen in blocks 500 and 502, respectively. If the TCP packet is an ACK packet, the attacker may be sending the attacking packets. Typically, the attacker's computer will wait about four minutes before being allowed to try and send another packet. In this case, if the packet is ignored as seen in a block 506, the attacker's efforts are effectively slowed while his computer waits over the appropriate response period.

[0124] FIGURES 26A and 26B represent a potential defense against an attack. In this method 510, the communications between an attacking computer and other devices on the local network is controlled. The system may accept a packet as seen in a block 512. The source IP address and the target IP address may be tested to determine whether they represent a threat as seen in blocks 514 and 520, respectively. If they do represent a threat, the target physical address may be tested to determine whether it is a synthetic address as seen in a block 516. If the target physical address is synthetic, the packet may be dropped as seen in a block 518 or forwarded to an alternate device or appropriate network security device such as a sacrificial computer or other defense device. The target physical address or synthetic address may be hardware addresses such as MAC addresses.

[0125] If the target physical address is not synthetic, then the system may be under attack from a new source and an appropriate control must be implemented. In this case, the IP address is tested to determine whether the source is on the local network as seen in a block 530. If the source is on the local network, then a single synthetic physical address is created. The synthetic physical address may, for example, be an address not in use on the local segment. Then, for each local device on the local segment, the system may create an ARP packet using the attacker's source IP address and the synthetic physical address as the source of the ARP packet as seen in a block 536. An example of the ARP packet is described in relation to FIGURE 19. The destination of the ARP packet may be the IP address and physical address of the local device as seen in a block 538. The packet may then be sent as seen in a block 540. This process is repeated for all devices on the local network. This method effectively adjusts the ARP tables on local devices, providing a false address for the attacker's computer.

[0126] Subsequently, for each device on a local network segment, a unique synthetic

physical address may be created in a process similar to that seen in a block 552. An ARP packet may then be created using the local device source IP address and the synthetic MAC address as the source of the ARP packet. The physical address of the threat may be set as the destination of the ARP packet similar to that seen in a block 556. Then, the ARP packet may be sent to the threat similar to that seen in a block 558. Again, this process is repeated for all devices on the local network. In effect, communications emanating from the threat entering are sent to addresses other than those of the devices on the local network, masking those devices.

[0127] If the source is not on the local network, then for each device on a local network segment, a unique synthetic physical address may be created as seen in a block 552. An ARP packet may then be created using the local device source IP address and the synthetic MAC address as the source of the ARP packet. The physical address of the default gateways may be set as the destination of the ARP packet as seen in a block 556. Then, the ARP packet may be sent to the default gateway as seen in a block 558. Again, this process is repeated for all devices on the local network. In effect, communications entering the local network are sent to addresses other than those of the devices on the local network, masking those devices from devices external to the network.

[0128] If however, the source IP address and the target IP address are not of interest or do not represent threats, the packet may be tested to determine whether the target physical address is synthetic as seen in a block 552. If the target address is synthetic, the system may replace the synthetic address with an appropriate real address of the target device as seen in a block 524. Subsequently, the system may send the reformulated packet as seen in a block 526.

[0129] In performing this method, the system may maintain a listing of synthetic hardware or physical addresses and the real hardware or physical address on the network. When checking for a synthetic address, the system may compare the hardware address to the list. When seeking to reform the packet, the system may substitute the real hardware or physical address for the synthetic address

[0130] FIGURE 27 is a schematic block diagram depicting an exemplary embodiment of a system. In this case, a local network segment is depicted to the right side of a router. Along the segment may be physical devices 1 and 2, and a security device D. There may also be

various physical addresses, S1, S2 and S3, that are not in use on the local network segment. If a threat penetrates the local network segment as denoted by the "T" block to the right of the router, then any communications with the threat outside the local network segment may be effectively controlled by preventing devices 1 and 2 from communicating with the threat. In this case, the ARP tables of these devices 1 and 2 may be adjusted such that they send their responses to a synthetic address on the network. In this manner, the threat inside the local network segment will never receive a reply to any messages sent to devices 1 and 2 on the local network segment. This disruption may be accomplished by a security device D sending an ARP packet to devices 1 and 2 providing them with a false physical address associated with the threat's IP address.

[0131] In another exemplary embodiment, the threat may exist outside the local network segment as denoted by the threat block. In this case, the threat may be mitigated by providing a false physical address for each device on the local network. In this case, any communications designed to go to device 1 or 2 may be instead be directed to a synthetic address such as S1 or S2. The security device D may send synthetic addresses for each device 1 and 2 on the network to a gateway device. In this manner, when the gateway device seeks to route communications from the threat outside the local network segment to devices on the network, it instead sends these packets to the synthetic address S1 and S2, respectively.

[0132] Aspects of the invention may be found in a system and method for tracking communication for determining device states. The method can include observing communication between devices and inferring a respective state of at least one of the devices based upon the communication observed. The inferring the respective state of a device can be performed without sending a packet to the device, without participating in the communication with the device, and only by listening to the communication with the device.

[0133] The method can further include setting a designation for a first device to a threat when the first device receives a packet and the respective state of the first device is unfulfilled. The method can further include changing the designation for the first device to a non-threat when subsequent communication initiated by the first device does not violate a rule for the communication.

[0134] The method can include setting a designation for a first device to a possible threat

when communication is initiated by the first device, and the communication initiated by the first device violates a rule. The method can further include changing the designation for the first device to a non-threat when subsequent communication initiated by the first device does not violate rules for the communication. The method can include setting a designation for a first device to a possible threat based upon a packet configuration for a packet sent by the first device as part of the communication.

[0135] Various states of the devices include unknown, used, unfulfilled, virtual, omitted, and automatic. The respective state of a device is determined to be unknown when the observation shows that the device fails to respond to communication sent to the first device. The respective state of the device is determined to be unfulfilled when an address resolution protocol request comprising a destination address for the device is observed, and the device does not respond to the address resolution protocol request prior to expiration of a time limit. The respective state of a device is determined to be used when the observation indicates that the device performs one of sending and receiving a packet. In addition, the respective state for the device is determined to be used when the observation shows that the device received a packet when its respective state was unfulfilled, and the device sent a reply to the packet within a time limit.

[0136] The respective state of a device is determined to be virtual when the observation shows that the device received a packet when its respective state was unfulfilled, and the device did not send a reply to the packet within a time limit. The respective state of the first device is determined to be automatic when an automatic reply is programmed to be sent to a second address when the first address receives a packet from the second address. The respective state of the device is determined to be omitted when the device has been programmed such that communication with the first address is omitted from observation.

[0137] In one embodiment, the method includes initializing the respective state of at least one device to unknown prior to beginning the observation. The plurality of devices can communicate via a segment of a network. In one embodiment, the method includes maintaining the respective state for one device in a storage area. In one embodiment, the method includes storing information about at least one packet of a plurality of packets communicated between the devices.

[0138] Additional aspects of the invention may be found in a system for tracking

communication for determining device states. The system may be a computational device including a processor or network interface and memory or computer-readable medium, among others. The device may or may not include a user interface. Further, the device may have various data and instructions associated with various methods for tracking .communication for determining device states These data may include an ARP request queue, an IP state table, a frequency table, an ARP table, a watch list, a threat list, a synthetic physical address table, and a communications stream table, among others. The device may also include instructions for evaluating packet reconnaissance rules, behavioral rules and other rules, among others. Further, the system may include software or computer interpretable instructions for performing various methods associated with maintaining the data in the tables and collecting the data for the tables.

[0139] As such, a system and method for tracking communication for determining device states is described. In view of the above detailed description of the present invention and associated drawings, other modifications and variations will now become apparent to those skilled in the art. It should also be apparent that such other modifications and variations may be effected without departing from the spirit and scope of the present invention as set forth in the claims which follow.